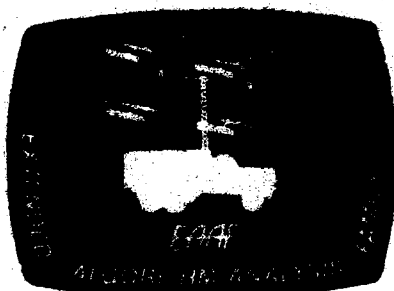AD-A194 563

7057-71

U.S. Army Intelligence Center and School Software Analysis and Management System

Minimization Of Squared Angular Error To Reduce Bias

Technical Memorandum No. 26

MARC

Mathematical Analysis Research Corporation

23 March 1987

National Aeronautics and
Space Administration

JPL

JET PROPULSION LABORATORY
California Institute of Technology
Pasadena, California

JPL D-4623
ALGO_PUB_0096

88   5 26 035

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER ALGO-PUB-0096	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER

4. TITLE *(and Subtitle)* Technical Memo 26, "Minimization of Squared Angular Error to Reduce Bias"	5. TYPE OF REPORT & PERIOD COVERED FINAL
	6. PERFORMING ORG. REPORT NUMBER D-4623

7. AUTHOR(s) Mathematical Analysis Research Corp. (MARC)	8. CONTRACT OR GRANT NUMBER(s) NAS7-918

9. PERFORMING ORGANIZATION NAME AND ADDRESS Jet Propulsion Laboratory, ATTN: 171-209 California Institute of Technology 4800 Oak Grove, Pasadena, CA 91109	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS RE 182 AMEND #187

11. CONTROLLING OFFICE NAME AND ADDRESS Commander, USAICS ATTN: ATSI-CD-SF Ft. Huachuca, AZ 85613-7000	12. REPORT DATE 23 Mar 87
	13. NUMBER OF PAGES 9

14. MONITORING AGENCY NAME & ADDRESS*(If different from Controlling Office)* Commander, USAICS ATTN: ATSI-CD-SF Ft. Huachuca, AZ 85613-7000	15. SECURITY CLASS. *(of this report)* UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE NONE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for Public Dissemination

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

Prepared by Jet Propulsion Laboratory for the US Army Intelligence Center and School's Combat Developer's Support Facility.

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Weighted Perpendicular, Minimization of Squared Angular Error and Squared Sine of Angular Error Algorithms, Variance, Bias, Computer Code, Iterations, Program Speed

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

As a follow-up to the results of "Two Dimensional Uncorrelated Bias in Fix Algorithms", DTIC #AD-A189473, MARC has examined the replacement of the Weighted Perpendicular algorithm withe the Minimization of Squared Angular Error algorithm. Standard modeled errors are discussed. Basic code for both the Minimization of Squared Angular Error and the Squared Sine of Angular Error are included.

DD FORM 1 JAN 73 **1473** EDITION OF 1 NOV 65 IS OBSOLETE

7057-71

U.S. ARMY INTELLIGENCE CENTER AND SCHOOL
Software Analysis and Management System
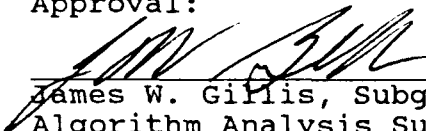

Minimization Of Squared Angular Error
To Reduced Bias

Technical Memorandum No. 26

23 March 1987

Author:

Michael W. Rennie

MARC
Mathematical Analysis Research Corporation

Approval:

James W. Gillis, Subgroup Leader
Algorithm Analysis Subgroup

Edward J. Records, Supervisor
USAMS Task

Concur:

A. F. Ellman, Manager
Ground Data Systems Section

Fred Vote, Manager
Advanced Tactical Systems

JET PROPULSION LABORATORY
California Institute of Technology
Pasadena, California

JPL D-4623

PREFACE

The work described in this publication was performed by the Mathematical Analysis Research Corporation (MARC) under contract to the Jet Propulsion Laboratory, an operating division of the California Institute of Technology. This activity is sponsored by the Jet Propulsion Laboratory under contract NAS7-918, RE182, A187 with the National Aeronautics and Space Administration, for the United States Army Intelligence Center and School.

This specific work was performed in accordance with the FY-87 statement of work (SOW #2).

iii

EXECUTIVE SUMMARY

This Technical Memorandum was prepared to summarize the results of work preformed under both the FY-86 and FY-87 Statements of Work and was funded by the FY-86 funds.

The purpose of this Technical Memorandum is to finalize the results of one phase of the statistical bias investigation for fixing algorithms. It discusses the trade-offs between the two competing optimization factors.

Minimization Of Squared Angular Error To Reduce Bias

SUMMARY

The systems reviewed to date use the weighted perpendicular method. As the number of bearings increases the bias of this approach becomes a larger and larger percentage of the total error. This does not happen to minimization of angular error. On the other hand, minimization of angular error is harder to compute. How much harder?

   I. Number of Iterations after a Perpendicular Method Start-
         Weighted Perpendicular converges in one or two iterations whereas Minimization of Squared Angular Error of angular error should probably use two to three iterations (after using the Perpendicular Method.)

  II. Shared calculations-
         In the Weighted Perpendicular reweighting can be done on saved terms without significantly reducing recalculation. Minimization of Squared Angular Error must recalculate all terms each iteration.

 III. Number of multiplications and additions-
         Minimization of Squared Angular Error has slightly more than Weighted Perpendicular. This difference should be fairly insignificant for total program run time however.

  IV. Computation of Transcendental Functions-
         Minimization of Squared Angular Error requires computing an inverse trig. All other methods require computation of trig functions but minimization of angular error does not. The primary difference is probably not the difference between trig functions and inverse trig functions. The primary difference is the trig functions need only be computed once whereas the inverse trig functions must be recomputed each time a new fix is found. To get the benefits of a fix with bias properties similar to Minimization of Squared Angular Error without having to recompute transcendental functions one should use Minimization of Sine Squared of Angular Error. According to its documentation this sin variation was used by FFIX. MARC has not seen the algorithm FFIX used. Our own code for this method is included later in this report.

      MARC recommends that slowdown in the number of fixes that can be processed be compared against the percentage of error resulting from fix algorithm bias. Analysis of data from field trials should be an important part of determining this percentage. Analysis based on models can predict what the result will be but only field trials can validate that important sources of error have not been left out of a model. Models should be a consideration in the planning and evaluation of trials but they are only one

element of an integrated investigation.

The remainder of this memo consists of:
  I. a section giving a general background on the issue of
     error from the fix algorithm (excluding bearing selection)
  II. a section giving the algorithm for one iteration of the
      minimization of squared angular error

I. FIX ALGORITHM ERRORS (EXCLUDING BEARING SELECTION ERROR)

Analysis of fix algorithms under the standard normal angular error model
yields the following conclusions:

A. Minimizing variance:
   Most of the error for small numbers of LOBs is reflected in the
   variance of the estimate. Minimization of the variance is therefore
   critical. Most fix methods have variances that are very similar.
   The method of minimization of Perpendicular Distance is an exception
   and hence should not be used. A Senior Thesis underway at Claremont
   McKenna College on a generalization of the Cramer Rao Inequality
   will show that the variance of the techniques such as Weighted
   Perpendicular, Minimization of Squared Angular Error, and Minimization
   of Sine Squared of Angular Error are essentially optimal relative to
   errors resulting from a shared angular uncertainty. (If there are known
   differences in angular uncertainty they should be incorporated. Any
   of these methods could be modified to weight for proportional accuracy.)
   The amount that the variance of the Perpendicular Method is larger
   depends on how much the relative distance from the emitter to the
   sensors varies depending on the sensor.

B. Reducing Bias:
   For small sample sizes bias is the second largest source of error.
   For large sample sizes it is the largest source of error. It can be
   shown that for most methods (certainly all Minimization of Square Error
   based methods) and 2 bearings the different methods give the same fix,
   and hence have the same bias. As sample size goes up some methods have
   bias shrink inversely proportionately to sample size and others do not.
   The Perpendicular Method and Weighted Perpendicular Method do not have
   shrinking bias in general. In fact, in some situations bias can
   increase with sample size for Perpendicular and Weighted Perpendicular.

C. Skewness:
   Intuition suggests that skewness depends on the relative size of an
   ellipse compared to the distance from the sensors. Skewness should
   decrease as sample size increases if this is the case. MARC would
   expect Weighted Perpendicular to have skewness similar to Minimization
   of Squared Angular Error. Furthermore MARC expects that both would be
   somewhat smaller than the skewness of the unweighted Perpendicular
   method in a way directly related to the larger variance of the
   Perpendicular method. MARC has not, however, tested these theories.

II. ITERATIONS TO UPDATE LOCATION ESTIMATES

Commented Squared Angular Error Minimization code is Attachment I.
Minimization of Sin Squared of Angular Error code is Attachment II.
Notes on the code: X and Y are used as subscripts. The initialization
portion of the program is not included but it sets X=1 (for the first
coordinate or X coordinate) and Y=2 (for the second coordinate or Y

coordinate.) ATN is the Arctangent function. Bearing(I) is the angle in degrees from true North of the bearing. The algorithm used is to update successive estimates of the fix by subtracting the inverse of the Hessian (second derivative) of the likelihood times the gradient of the likelihood.

----- ALGORITHM LISTING -----
ITERATION FROM MINIMIZATION OF ANGULAR ERROR

```
2548      ! ******************** Variables *************************
2549      !
2550      ! Partial_wrt(X) is the partial of the likelihood L with respect to X
2551      ! Partial_wrt(Y) is the partial of the likelihood L with respect to Y
2552      ! L(X,X),L(X,Y), and L(Y,Y) are the second partials of L
2553      ! Diff=Angular Difference
2554      ! Sensor(I,X)=X coordinate of the ith sensor
2555      ! Sensor(I,Y)=Y coordinate of the ith sensor
2556      ! Fix(X)=X coordinate of the current location estimate
2557      ! Fix(Y)=Y coordinate of the current location estimate
2558      ! Dx is the difference in X coordinates of the estimate and the sensor
2559      ! Dy is the difference in Y coordinates of the estimate and the sensor
2560      ! Determinant=L(X,X)L(Y,Y)-L(X,Y)L(Y,X)
2561      ! Other variables should be self-explanatory
2562      !
2563      ! ENTER THIS LOOP WITH AN ESTIMATE BASED ON THE PERPENDICULAR METHOD
2564      !
2565  FOR Iteration=1 TO 4
2566      ! **************** INITIALIZE TO COMPUTE A SUMMATION ***************
2567      Partial_wrt(X)=0
2568      Partial_wrt(Y)=0
2569      L(X,X)=0
2570      L(X,Y)=0
2571      L(Y,Y)=0
2572      ! ************* COMPUTE SUMMATION FOR L & Partial_wrt **************
2573      FOR I=1 TO Nsensors
2574        Diff=(Bearing(I)-ATN((Sensor(I,X)-Fix(X))/(Sensor(I,Y)-Fix(Y))))
2575        IF Diff>180 THEN Diff=Diff-360
2576        IF Diff<-180 THEN Diff=Diff+360
2577        Diff=Diff*PI/180.0                    ! *** convert to radians ***
2578        Dx=Fix(X)-Sensor(I,X)
2579        Dy=Fix(Y)-Sensor(I,Y)
2580        Dx_2=Dx*Dx
2581        Dy_2=Dy*Dy
2582        Dxy=Dx*Dy
2584        Distance_sq=Dx_2+Dy_2
2585        Distance_4=Distance_sq*Distance_sq
2586        Partial_wrt(X)=Partial_wrt(X)-2.0*Dy*Diff/Distance_sq
2587        Partial_wrt(Y)=Partial_wrt(Y)+2.0*Dx*Diff/Distance_sq
2588        L(X,X)=L(X,X)+2.0*(2.0*Diff*Dxy+Dy_2)/Distance_4
2589        L(X,Y)=L(X,Y)+2.0*((2.0*Diff*Dy_2-Dxy)/Distance_4-Diff/Distance_sq)
2590        L(Y,Y)=L(Y,Y)+2.0*(-2.0*Diff*Dxy+Dx_2)/Distance_4
2591      NEXT I
2592      L(Y,X)=L(X,Y)              ! *** no sense computing it twice ***
2593      !
2594      ! Now update the Fix by subtracting (L inverse) times (Partial_wrt)
2595      !
2596      Determinant=L(X,X)*L(Y,Y)-L(X,Y)*L(Y,X)
2597      Fix(X)=Fix(X)-(L(Y,Y)*Partial_wrt(X)-L(X,Y)*Partial_wrt(Y))/Determinant
2598      Fix(Y)=Fix(Y)-(-L(Y,X)*Partial_wrt(X)+L(X,X)*Partial_wrt(Y))/Determinant
2599      PRINT "FIX=(";Fix(X);",";Fix(Y);")"
2600  NEXT Iteration
```

—— ALGORITHM LISTING ——
ITERATION FROM MINIMIZATION OF SIN SQUARED OF ANGULAR ERROR

```
2629  FOR I=1 TO Nsensors
2630    S2(I)=SIN(2*Bearing(I))
2631    C2(I)=COS(2*Bearing(I))
2641  NEXT I
2653  FOR Iteration=1 TO 3
2654    ! *************** INITIALIZE TO COMPUTE A SUMMATION ***************
2655    Partial_wrt(X)=0
2656    Partial_wrt(Y)=0
2657    L(X,X)=0
2658    L(X,Y)=0
2659    L(Y,Y)=0
2662    ! ************* COMPUTE SUMMATION FOR L & Partial_wrt **************
2663    FOR I=1 TO Nsensors
2666      Dx=Fix(X)-Sensor(I,X)
2667      Dy=Fix(Y)-Sensor(I,Y)
2668      Dx_2=Dx*Dx
2669      Dy_2=Dy*Dy
2670      Dx_3=Dx_2*Dx
2671      Dy_3=Dy_2*Dy
2672      Dx_4=Dx_3*Dx
2673      Dy_4=Dy_3*Dy
2674      Dxy=Dx*Dy
2676      Distance_sq=Dx_2+Dy_2
2677      D4=Distance_sq*Distance_sq
2678      D6=Distance_4*Distance_sq
2679      S=S2(I)
2680      C=C2(I)
2681      Partial_wrt(X)=Partial_wrt(X)+Dy*(S*(Dx_2-Dy_2)+2.0*C*Dxy)/D4
2682      Partial_wrt(Y)=Partial_wrt(Y)+Dx*(S*(Dy_2-Dx_2)-2.0*C*Dxy)/D4
2683      L(X,X)=L(X,X)+2.0*(S*(3.0*Dx*Dy_3-Dx_3*Dy)-C*(3.0*Dx_2*Dy_2-Dy_4))/D6
2684      L(X,Y)=L(X,Y)+(S*(Dx_4-6*Dx_2*Dy_2+Dy_4)+4*C*(Dx_3*Dy-Dx*Dy_3))/D6
2685      L(Y,Y)=L(Y,Y)+2.0*(S*(3.0*Dx_3*Dy-Dx*Dy_3)+C*(3.0*Dx_2*Dy_2-Dx_4))/D6
2686    NEXT I
2687    L(Y,X)=L(X,Y)                  ! *** no sense computing it twice ***
2688    !
2689    ! Now update the Fix by subtracting (L inverse) times (Partial_wrt)
2690    !
2691    Determinant=L(X,X)*L(Y,Y)-L(X,Y)*L(Y,X)
2692    Fix(X)=Fix(X)-(L(Y,Y)*Partial_wrt(X)-L(X,Y)*Partial_wrt(Y))/Determinant
2693    Fix(Y)=Fix(Y)-(-L(Y,X)*Partial_wrt(X)+L(X,X)*Partial_wrt(Y))/Determinant
2694    PRINT "FIX=(";Fix(X);",";Fix(Y);")"
2695  NEXT Iteration
```